

Challenges in Bioinformatics Lab: Distributable Packages

Ariane C. Boehm

Challenges

- Make Challenge
- Programing Challenge
 - Python
 - C++
- Details

Challenges

- Make Challenge
- Programing Challenge
 - Python
 - C++
- Details

having
fun with
GNU
and other
Docu's!



Make Challenge

```
#!/usr/bin/env make
all: crCUT crCHK crXXD allProg.sh

crCUT: cutProg.sh
    bash cutProg.sh

crXXD: xxdProg.sh
    bash xxdProg.sh

crCHK: chkProg.sh
    bash chkProg.sh

clean:
    rm -r *.xxd
    rm -r *.chk
    rm first.name
    rm last.name

help:
    echo "./Makefile-ACB [Args]"
    echo "possible Args:"
    echo "[crCUT] creates two files: first/last.name out of full.name"
    echo "[crXXD] creates the xxdFile"
    echo "[crCHK] creates the CHKFile"
    echo "[clean] removes all created files"
    echo "[all] does crCUT, crXXD and crCHK and displays the content of these files"

.PHONY: clean help
~
```

Make Challenge

```
#!/usr/bin/make -f
NAMES=first.name last.name full.name
XNAMES=$(NAMES:.name=.xxd)
CNAMES=$(NAMES:.name=.chk)
CUT=cut -d' '

all: first.name last.name $(CNAMES) $(XNAMES)
    cat *.xxd *.chk

first.name: full.name
    $(CUT) -f1 full.name > first.name

last.name: full.name
    $(CUT) -f2 full.name > last.name

$(CNAMES): %.chk:%.name
    shasum $< > $@

$(XNAMES): %.xxd:%.name
    xxd $< $@

clean:
    rm -f *.xxd *.chk first.name last.name

help:
    echo "./Makefile-ACB [Args]"
    echo "possible Args:"
    echo "[first.name] creates the first.name file (also possible: last)"
    echo "[first.xxd] creates the xxdFile of first.name (also possible: last, full)"
    echo "[first.chk] creates the CHKFile of first.name (also possible: last, full)"
    echo "[clean] removes all created files"
    echo "[all] does CUT, XXD and CHK and displays the content of these files"

.PHONY: clean help
```

Make Challenge

Shell Command Line:

```
make -f Makefile [ARGS]
```

Python Challenge

removeSpaces.py

```
#####  
# Method removeSpaces  
#####  
def removeSpaces(infile):  
    outfilename = "returntext.txt"  
  
    infile = open(infile, 'r')  
    outfile = open(outfilename, 'w')  
  
    for line in infile:  
        tmp = line.replace(" ", "")  
        outfile.write(tmp)  
  
    infile.close()  
    outfile.close()
```

definition of method (parameter - infile)
outfilename

opens the infile (r = read)
opens the outfile (w = write)

for each line in my infile
replace the space by "" in an temporary string
write the temp sting directly to the outfile

close infile
close outfile

Python Challenge

setup.py

```
from distutils.core import setup
setup(name='removeSpaces',
      version='1.0',
      description='A program that removes whitespaces from a given textfile',
      author='Ariane Boehm',
      author_email='boehma@in.tum.de',
      url='ariane@ariane-boehm.com',
      py_modules=['removeSpaces'],
      )
```

Python Challenge

Shell Command Line:

```
python setup.py sdist
```

C++ Challenge

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

// Main help:  cplusplus.com - Tutorial!
int main () {
    string line;
    ifstream infile ("example.txt");
    ofstream outfile;
    outfile.open ("returntext.txt");
    if (infile.is_open()){
        while ( infile.good() ){
            getline (infile,line);
            for(int i = 0; i < line.length(); i++){
                if(line[i] == ' '){
                }
                else{
                    if (outfile.is_open()){
                        outfile << line[i];
                    }
                    // if outfile can't be opened - print an error in the terminal
                    else cout << "Error opening the output file";
                }
            }
        }
        outfile.close();
        infile.close();
    }
    else cout << "Unable to open file";

    return 0;
}

// input and output
// file streaming

// initialize string line
// InFile Stream of example.txt
// OutFile Stream of outfile
// opens the outfile returntext.txt
// if the file can be opened - open it
// until end of file
// gets the line of infile
// look for whitespaces
// (I did not find a replace Method...)

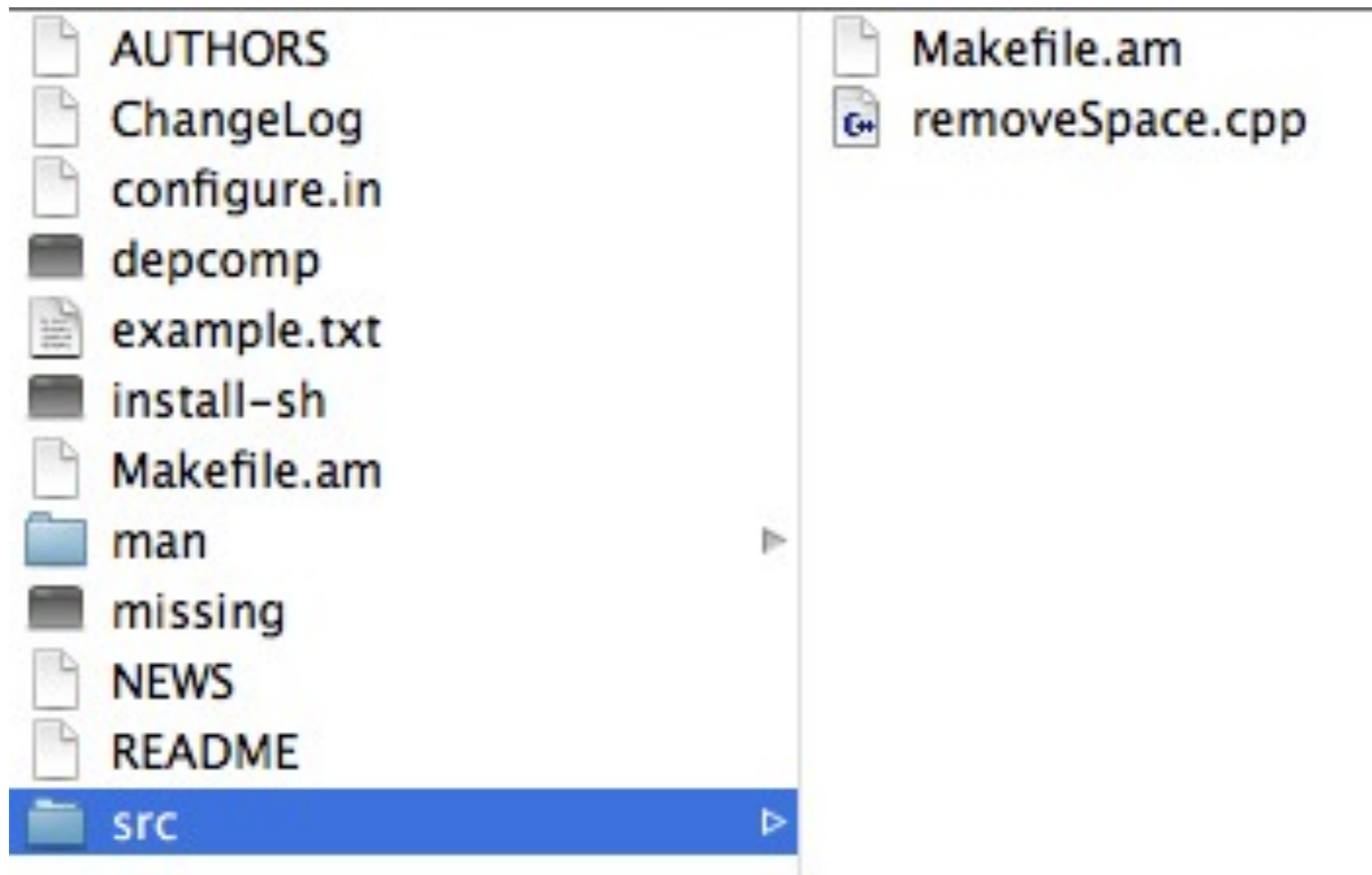
//checking if the outfile is already open
// print this part of the line in the outfile

// closes the outfile
// close the input file

// else c-out: Unable to open the file

// on success (1 describes an error)
```

C++ Challenge

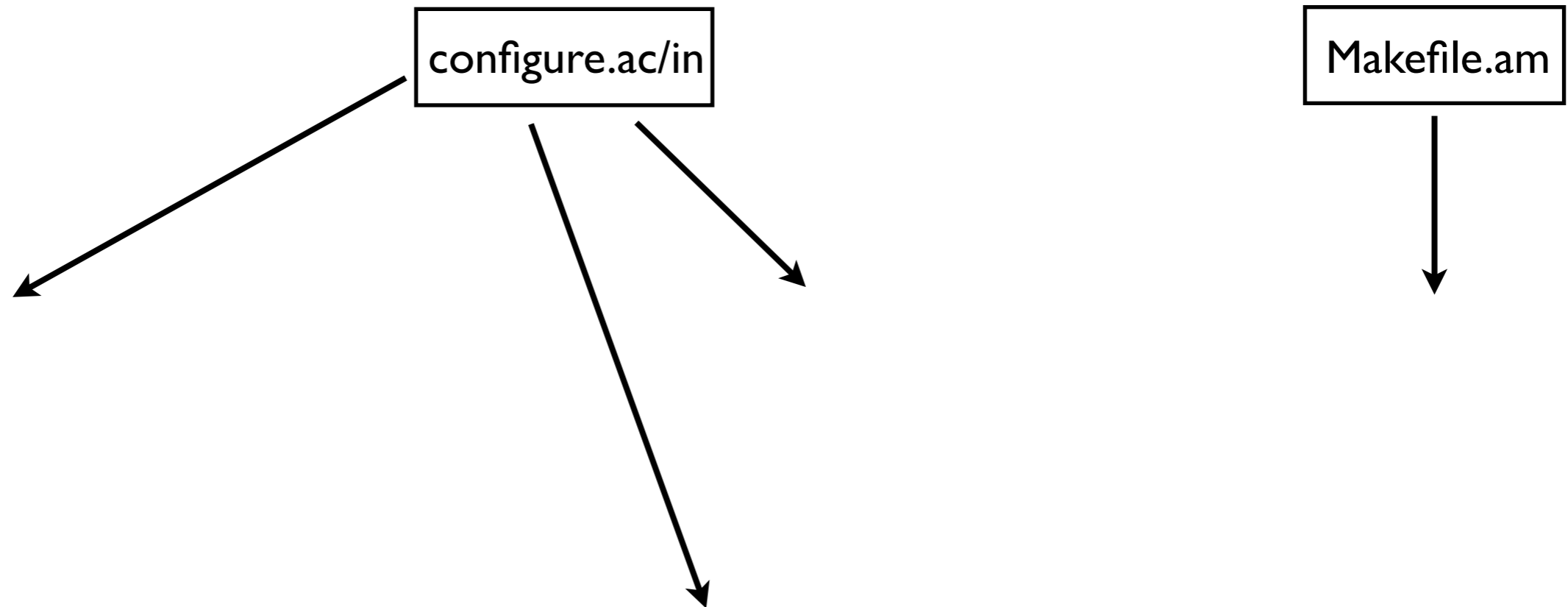


C++ Challenge

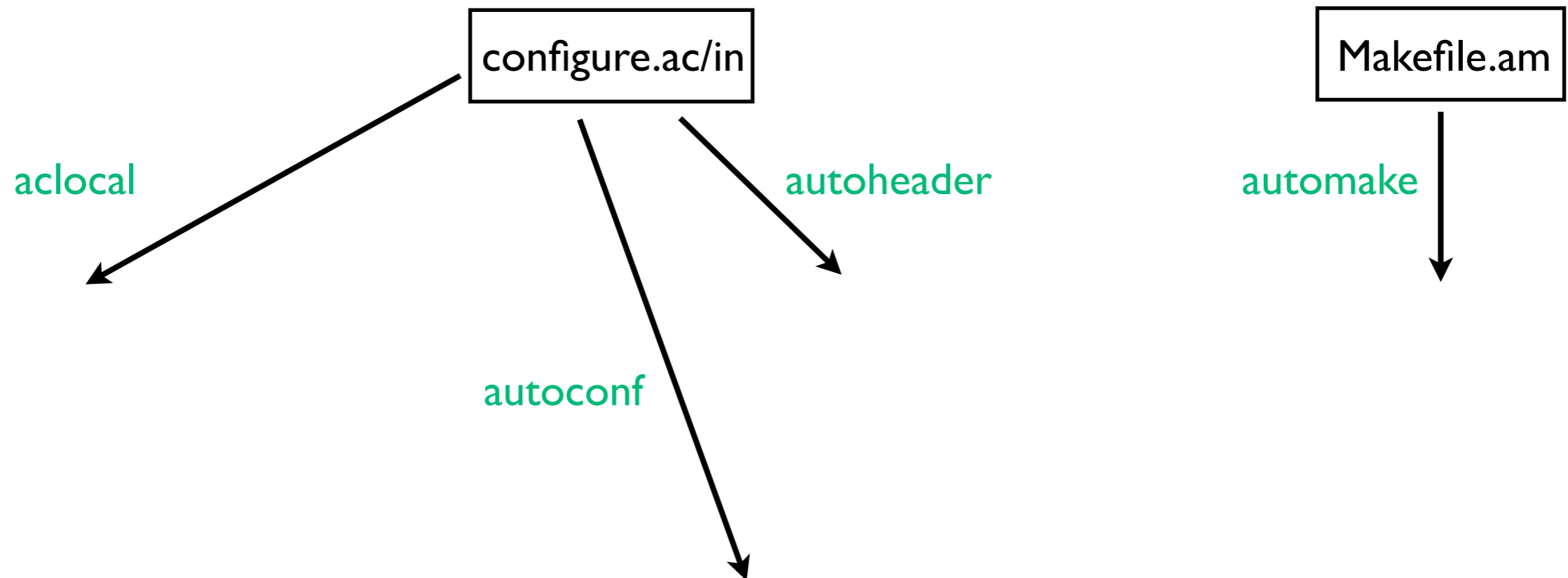
configure.ac/in

Makefile.am

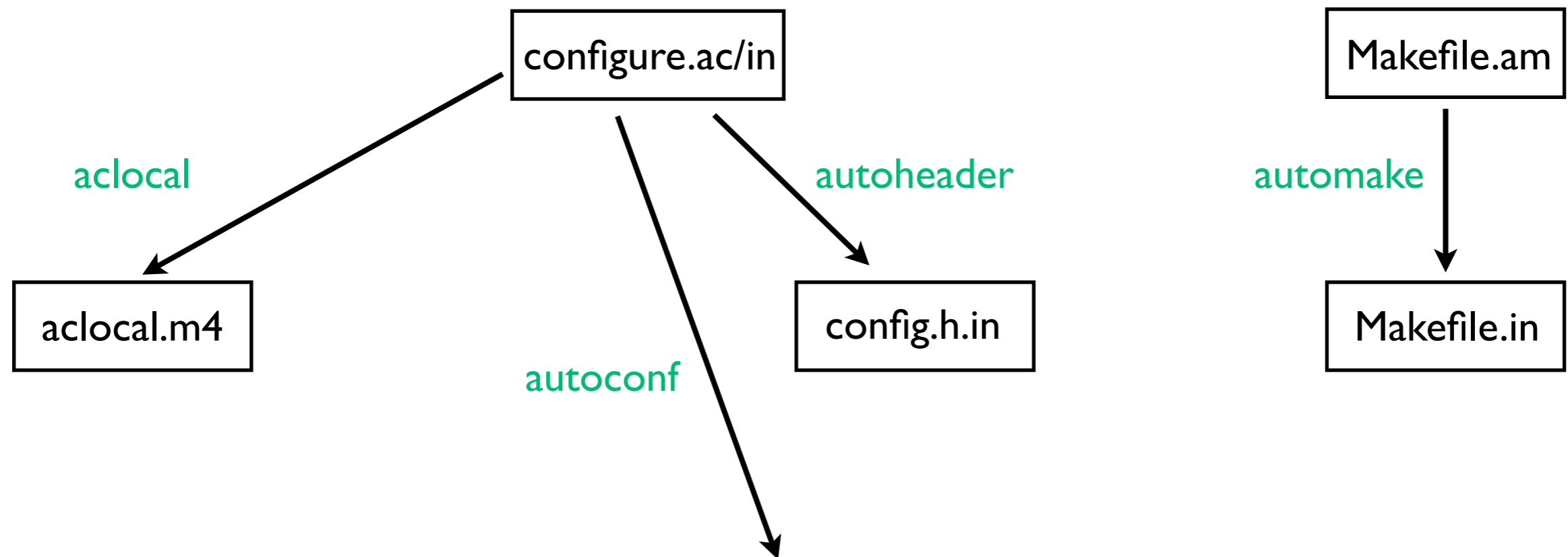
C++ Challenge



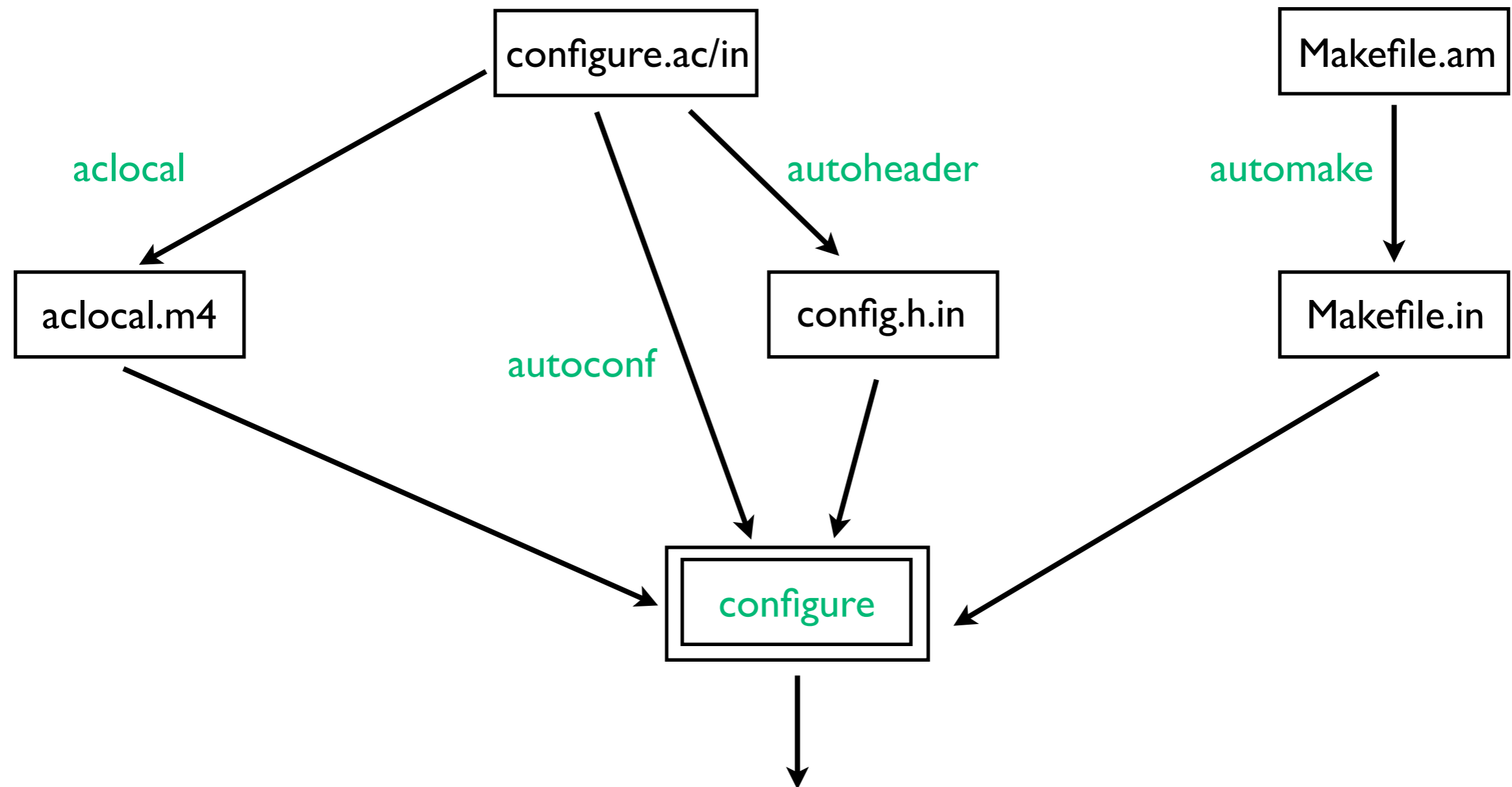
C++ Challenge



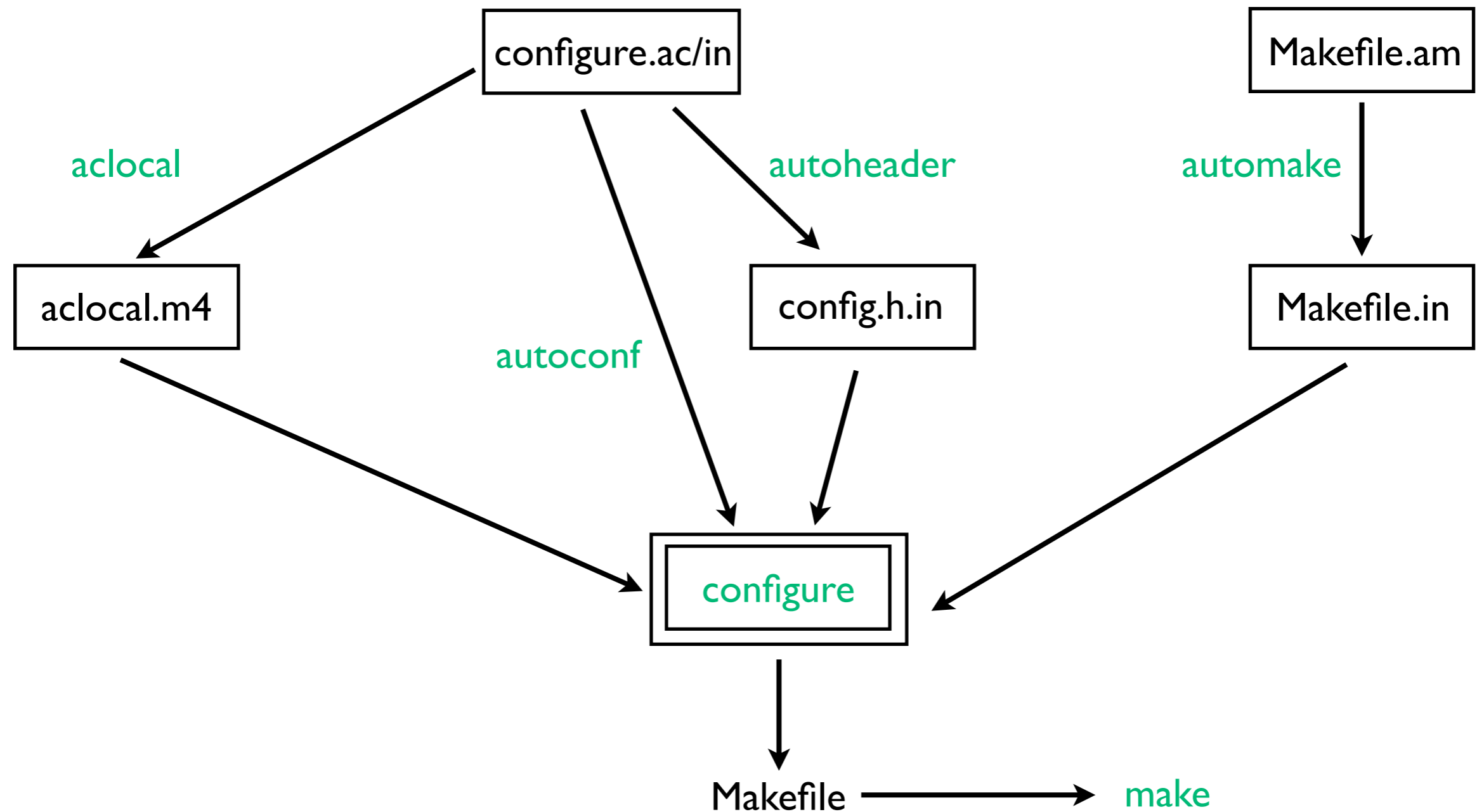
C++ Challenge



C++ Challenge



C++ Challenge



C++ Challenge

Shell Command Lines:

aclocal

autoheader

autoconf

automake



./configure

make

make install

make distcheck

C++ Challenge

Shell Command Lines:

aclocal
autoheader
autoconf
automake

./configure

make

make install

make distcheck



```
=====
removeSpace-0.1 archives ready for distribution:
removeSpace-0.1.tar.gz
=====
```

C++ Challenge

Shell Command Lines:

aclocal
autoheader
autoconf
automake

./configure

make
make install
make distcheck



```
aclocal.m4  
AUTHORS  
autom4te.cache  
ChangeLog  
config.h  
config.h.in  
config.log  
config.status  
configure  
configure.in  
depcomp  
example.txt  
install-sh  
Makefile  
Makefile.am  
Makefile.in  
man  
missing  
NEWS  
README  
removeSpace-0.1.tar.gz  
src  
stamp-h1
```

Details

Python

```
#!/usr/bin/python
```

```
"""
```

```
tralalala
```

```
This module removes whitespaces from a given textfile. Usage ./removeSpaces infile!
```

```
"""
```

```
__author__ = 'Ariane Boehm'
```

```
__version__ = '1.0'
```

```
import sys
```

```
#####
```

```
# Main Method; nothing to say here
```

```
#####
```

```
def main():
```

```
    progname = sys.argv[0]
```

```
    if len(sys.argv) > 2:
```

```
        sys.exit("Usage: %s INFILE" %progname)
```

```
    try:
```

```
        infile = sys.argv[1]
```

```
    except IndexError:
```

```
        sys.exit("Usage: %s INFILE" %progname)
```

```
    removeSpaces(infile)
```

```
# name of the program
```

```
# check for length of input
```

```
# says the user how to use it
```

```
# set 1st parameter: infile
```

```
# if it doesn't work - (i.e.) no parameter
```

```
# says the user how to use it
```

```
# call method removeSpaces
```

Python

```
#!/usr/bin/python
```

```
"""
```

```
tralalala
```

```
This module removes whitespaces from a given textfile. Usage ./removeSpaces infile!
```

```
"""
```

```
__author__ = 'Ariane Boehm'
```

```
__version__ = '1.0'
```

```
import sys
```

```
#####
```

```
# Main Method; nothing to say here
```

```
#####
```

```
def main():
```

```
    progname = sys.argv[0]
```

```
    if len(sys.argv) > 2:
```

```
        sys.exit("Usage: %s INFILE" %progname)
```

```
    try:
```

```
        infile = sys.argv[1]
```

```
    except IndexError:
```

```
        sys.exit("Usage: %s INFILE" %progname)
```

```
    removeSpaces(infile)
```

```
# name of the program
```

```
# check for length of input
```

```
# says the user how to use it
```

```
# set 1st parameter: infile
```

```
# if it doesn't work - (i.e.) no parameter
```

```
# says the user how to use it
```

```
# call method removeSpaces
```

Python

Shell Command Line:

```
pydoc removeSpaces
```

```
python
```

```
>>import pydoc
```

```
>>import removeSpaces
```

```
>>pydoc.help(removeSpaces)
```

Help on module removeSpaces:

NAME

removeSpaces - tralalala

FILE

/Users/Ariane/Desktop/Ariane_Challenges/progChallengePY/removeSpaces.py

DESCRIPTION

This module removes whitespaces from a given textfile. Usage ./removeSpaces infile!

FUNCTIONS

main()

#####

Main Method; nothing to say here

#####

removeSpaces(infile)

#####

removeSpaces (is called by main)

#####

DATA

__author__ = 'Ariane Boehm'

__version__ = '1.0'

VERSION

1.0

AUTHOR

Ariane Boehm

~

Python

```
##### JUST FOR THE INTERPRETER
```

```
if __name__ == '__main__':
```

```
    try:
```

```
        main()
```

```
    except KeyboardInterrupt:
```

```
        pass
```

```
# if the python interpreter is running that module
```

```
# it sets the special __name__ variable
```

```
# to have a value "__main__".
```

```
# If this file is being imported from another module,
```

```
# __name__ will be set to a different value.
```

```
# tries to execute main
```

```
# except if the user interrupts the process (ctrl - c)
```

Thank you.